Compaction of Diagnostic Test Set for a Full-Response Dictionary[*]

Mohammed Ashfaq Shukoor[†] and Vishwani D. Agrawal
Auburn University, Department of Electrical and Computer Engineering, Auburn, AL 36849, USA
shukoma@auburn.edu, *vagrawal@eng.auburn.edu*

Abstract

We optimize the diagnostic data from a full-response fault dictionary of a given test set. Compaction is done without loss of diagnostic resolution of a test set. We give an integer linear program (ILP) formulation using fault diagnostic table. The complexity of the ILP is made manageable by two innovations. First, we define a generalized independence relation between pairs of faults to reduce the number of fault pairs that need to be distinguished. This significantly reduces the number of ILP constraints. Second, we propose a two-phase ILP approach. An initial ILP phase, which uses existing procedures, selects a minimal detection test set. In a final phase, additional tests are then selected for the undiagnosed faults using a new diagnostic ILP. The overall minimized test set may be only slightly longer than a one-step ILP optimization, but has advantages of significantly reduced computation complexity and reduced test time. Benchmark results show potential for very small diagnostic test sets.

Keywords - Fault diagnosis, integer linear programming, generalized independence, fault dictionary, test minimization.

1. Introduction

The process of determining the cause of failure of a chip is known as *failure analysis.* Failure analysis often leads to improvement in the design of the chip and/or the manufacturing process. *Fault diagnosis* is the first step in failure analysis which by logical analysis gives a list of likely defect sites or regions. Basically, fault diagnosis narrows down the area of the chip on which physical examination needs to be done to locate defects.

Diagnosis algorithms are broadly classified into two types: *effect-cause fault diagnosis* and *cause-effect fault diagnosis.* As the name suggests the effect-cause algorithm directly examines the syndrome of the failing chip and then derives the fault candidates [1] using path-tracing algorithms. The fault candidate here usually is a logical location or area of the chip.

On the other hand, the cause-effect algorithm starts with a particular fault model and compares the signature of the observed faulty behavior with the simulated signatures for each fault in the circuit. A *fault signature (or syndrome)* is a list of failing vectors and the outputs at which errors are detected. A cause-effect algorithm can further be classified as static, in which all fault simulation is done in advance and all fault signatures are stored as a fault dictionary or, as dynamic, where simulations are performed only as needed during the diagnosis process. As the cause-effect algorithms are based on a fault model and real defects on the chip may not behave according to the fault model used, the observed signature may not match with any of the simulated signatures. In such cases sophisticated techniques are used to select a set of signatures that best match the observed signature [11].

Despite its overwhelming data requirements, the fault dictionary based diagnosis has been popular as it facilitates faster diagnosis by comparing the observed behaviors with pre-computed signatures in the dictionary [4]. The most detailed form of fault dictionary which can provide all the information for a given test set is the *full-response dictionary.* It consists of all output responses of each fault for each test. On the other hand the most compact form of fault dictionary is a *pass-fail dictionary* which stores a single pass or fail bit for a fault-vector pair. The disadvantage with pass-fail dictionaries is that since the failing output information is ignored, faults that fail same set of tests but at different outputs cannot be distinguished [12]. Thus pass-fail dictionaries are not commonly used for fault diagnosis.

There has been a lot of work done to reduce the size of the full-response dictionary [5, 12, 15]. Most of these techniques concentrate on reducing the size by managing the organization and encoding of the

---

[*] This research is supported in part by the National Science Foundation Grant CNS-0708962.

[†] Presently with Texas Instruments, Dallas, TX 75243, Email: m-shukoor@ti.com

dictionary. Dictionary organization is the order and content of the information, and dictionary encoding is the data representation format in the dictionary. Very little work has been done on reducing the size of the dictionary by compaction of the diagnostic test set [8]. In this work we explore the idea of using a minimal test set for fault diagnosis.

We give an integer linear program (ILP) formulation to minimize test sets for a full-response dictionary based diagnosis. The ILP solution is a test set with diagnostic characteristics identical to that of the original unoptimized test set. Having a smaller test set not only reduces the dictionary size, but also reduces the time for debugging the faulty chip. An ideal test set for diagnosis is one which distinguishes all faults. Thus during diagnostic test set minimization it should be ensured that the resulting test set consists of at least one vector to distinguish every pair of faults. Notice that the number of fault pairs is proportional to the square of the number of faults. This results in a very large number of constraints in the ILP. We define a new diagnostic fault independence relation to reduce the number of fault pairs to be considered. Finally a two-phase method is proposed for generating a minimal diagnostic test set from any given test set. In the first phase we use existing ILP minimization techniques [17] to obtain a minimal detection test set and find the faults not diagnosed by this test set. In the second phase we use the diagnostic ILP to select a minimal set of vectors capable of diagnosing the undiagnosed faults from Phase-1. The resulting minimized test set combined with the minimal detection test set of Phase-1 serves as our complete diagnostic test set.

The rest of the paper is organized as follows. Section 2 gives the diagnostic ILP formulation and illustrates its complexity. Section 3 introduces a new diagnostic fault independence relation to reduce the number of constraints in the diagnostic ILP. Section 4 describes the two-phase method for generating a minimal diagnostic test set. Section 5 gives the results and Section 6 gives the conclusion.

2. ILP for Diagnostic Test Set Minimization

Integer linear programming (ILP) is an effective mathematical method for test optimization. It gives global optimization and has been used for both combinational and sequential circuits [6, 7] as well as for minimizing N-detect tests [9]. In [17], a primal-dual ILP algorithm is given for generating minimal detection test sets based on identifying independent faults, generating tests for them, and minimizing the tests. All of these ILP formulations use a fault detection table which contains information about faults detected by each vector. The fault detection table is obtained by fault simulation without fault dropping. Note that the information in a fault detection table is similar to that in the pass-fail dictionary.

	t1		t2		t3		t4		t5	
	o1	o2	o1	o2	o1	o2	o1	o2	o1	o2
f1	1	0	1	0	1	0	1	0	0	0
f2	1	1	1	1	1	0	1	1	0	0
f3	1	1	1	1	1	0	0	0	0	0
f4	0	1	0	1	0	0	0	1	0	0
f5	0	0	0	0	0	1	0	0	1	1
f6	0	0	0	0	0	1	0	0	0	0
f7	0	0	0	0	0	1	0	0	0	1
f8	0	0	1	0	1	0	1	0	0	0

Figure 1. Full-response fault dictionary.

	t1	t2	t3	t4	t5
f1	1	1	1	1	0
f2	2	2	1	2	0
f3	2	2	1	0	0
f4	3	3	0	3	0
f5	0	0	2	0	1
f6	0	0	2	0	0
f7	0	0	2	0	2
f8	0	1	1	1	0

Figure 2. Fault diagnostic table.

2.1. Fault Diagnostic Table for Diagnostic ILP

The ILP formulation for minimizing test sets used for full-response dictionary based diagnosis requires a matrix representation that not only tells which tests detect which faults, but also at which outputs the discrepancies were observed for each fault-test pair. For this reason we use a *fault diagnostic table*. We illustrate the construction of a fault diagnostic table with the following example.

Let us consider a circuit with 2 outputs, having 8 faults detected by 5 test vectors. A sample full response dictionary for such a circuit is shown in the Figure 1. Here '0' stands for pass and '1' stands for fail.

We use integers to represent the output response for each test vector. As faults detected by different test vectors are already distinguished, there is no need to compare the corresponding output responses. Hence we assign indices for the failing output responses for each test vector. In the example, for test t1 the 3 different failing output responses ("10", "11", and "01") are indexed by integers 1, 2 and 3 respectively in the fault diagnostic table as shown in Figure 2. The largest integer needed to index an output response in the worst case is equal to *minimum*($2^{No.\ of\ output\ pins}-1$, *highest number of faults detected by any test vector*). However it should be noted that output responses to a particular vector are likely to repeat across a fault set as faults in the same output cone can have identical output responses for a particular test. For this reason the largest integer needed to index an output response observed in our experiments was much smaller than the highest number of faults detected by any test vector.

2.2. Diagnostic ILP Formulation

Suppose a combinational circuit has K faults. We are given a vector set V of J vectors and we assign a [0, 1] integer variable v_j , $j = 1, 2, \dots,$ J to each vector. The variables v_j have the following meaning: If $v_j = 1$, then vector j is included in the selected vector set. If $v_j = 0$, then vector j is discarded.

Without loss of generality, we assume that all K faults are detected by vector set V and are also distinguishable from each other. Our problem then is to find the smallest subset of these vectors that distinguish all the fault pairs. We simulate the fault set and the vector set without dropping faults and the fault diagnostic table is constructed as explained in the previous section. In this table, an element $a_{kj} \geq 1$ only if fault k is detected by vector j. The diagnostic ILP problem is stated as,

$$\text{Minimize} \sum_{j=1}^{J} v_j \tag{1}$$

subject to,

$$\sum_{j=1}^{J} v_j\ a_{ij} \geq 1; \quad for\ i = 1, 2, ..., K \tag{2}$$

$$\sum_{j=1}^{J} v_j\ |a_{kj} - a_{pj}| \geq 1 \tag{3}$$

for, $k = 1, 2, .., K-1$ and $p = k+1, .., K$

$$v_j \in integer[0, 1], \quad j = 1, ..., J \tag{4}$$

The constraint set given by (2) consists of K constraints - called *detection constraints* which ensure that every fault is detected by at least one vector. The constraint set given by (3) consists of $K(K-1)/2$ constraints - one constraint for every fault pair. These are called the *diagnostic constraints.* A diagnostic constraint consists of vector variables corresponding to non-zero $|a_{kj} - a_{pj}|$, i.e., the vectors that produce different output responses for the k^{th} and p^{th} faults. It allows at least one of those vectors to be selected since the inequality is greater than or equal to 1. Thus the diagnostic constraint set insures that k^{th} fault is distinguished from the p^{th} fault by at least one vector in the selected vector set. Additionally, the provable ability of the ILP to find the optimum provided its execution is allowed to complete guarantees the smallest size test set. Note that the total number of constraints here is $K(K+1)/2$, which is proportional to the square of the number of faults.

3. Generalized Fault Independence

One clear disadvantage of the diagnostic ILP is that the number of constraints is a quadratic function of the number of faults. Thus, for large circuits the number of constraints would be unmanageable. To overcome this, we define a relation between a pair of faults which allows us to drop the diagnostic constraints in the ILP corresponding to many fault pairs. We have generalized the conventional *fault independence relation* given in the literature by considering the detection of faults at different primary outputs and relative to a vector set. Conventionally [3], a pair of faults is called *independent* if the faults are not detected by any common vector. This definition does not account for the detection of the faults at specific outputs. Also, it implies "absolute" independence, which is with respect to the exhaustive vector set. We generalize the definition of fault independence by saying that two faults detected by the same vector can still be called independent, provided the output responses of the two faults to that vector are different.

Definition: Generalized Fault Independence - A pair of faults detectable by a vector set V are said to be diagnostically independent with respect to vector set V, if there is no single vector that detects both the faults and produces an identical output response.

Note that the generalized independence relation is conditional to a vector set. Thus, the conventional independence can be viewed as a special case of the generalized independence, for a single output circuit and conditional to an exhaustive vector set.

	t1	t2	t3	t4
f1	1	0	1	0
f2	1	1	0	0
f3	0	0	1	1

Figure 3. Fault detection table.

Table 1. Independence relation.

Fault pair	Independence relation	Reason
f1, f2	NO	Both faults detected by t1
f1, f3	NO	Both faults detected by t3
f2, f3	YES	No vector detects both faults

Example: Consider a fault detection table with 3 faults and 4 test vectors as shown in Figure 3. The independence relation between every fault pair is given in Table 1.

Now consider a fault diagnosis table for the same set of faults and vectors as shown in Figure 4. Recall that the fault diagnosis table takes in to account the output responses for each fault-vector pair. It is constructed as explained in Section 2.1. The generalized independence relations for all pairs of faults are given in Table 2.

In the context of the diagnostic ILP, the diagnostic independence relation plays an important role in reducing the number of constraints to be used in the formulation. When two faults are diagnostically independent, any vector that detects either of the faults will be a distinguishing vector. Thus, in set (3), a constraint for a diagnostic independent fault pair will have vector variables corresponding to all the vectors that detect any one or both the faults. In the presence of detection constraints of set (2) which guarantee a test for every fault, a diagnostic constraint for an independent fault pair is redundant. Also, such a constraint will be covered by other diagnostic constraints corresponding to non-independent fault pairs containing a fault from the diagnostic independent fault pair.

The graph in Figure 5 shows the reduction in the constraint set sizes by considering diagnostic independent faults for a 4 bit ALU and few ISCAS85 benchmark circuits.

It can be seen that there is an order of magnitude reduction in the constraint set sizes on eliminating constraints corresponding to diagnostic independent faults. However the constraint set sizes still are large and need to be reduced to manageable proportions.

	t1	t2	t3	t4
f1	1	0	1	0
f2	2	1	0	0
f3	0	0	1	1

Figure 4. Fault diagnostic table.

Table 2. Generalized independence relation.

Fault pair	Generalized indep. relation	Reason
f1, f2	YES	Different output responses for t1 detecting both faults
f1, f3	NO	Identical output responses for t3 detecting both faults
f2, f3	YES	No vector detects both faults

4. Two-Phase Minimization

Given an unoptimized test set, we proceed as [16]:

Phase 1: *Use existing ILP minimization techniques [17] to obtain a minimal detection test set from the given unoptimized test set. Find the faults not diagnosed by the minimized test set.*

Phase 2: *Run the diagnostic ILP on the remaining unoptimized test set to obtain a minimal set of vectors to diagnose the undistinguished faults from Phase-1. The resulting minimized test set combined with the minimal detection test set of Phase-1 serves as a complete diagnostic test set.*

In the context of diagnostic ILP of Phase-2, the Phase-1 along with the generalized independence relation helps in reducing the number of constraints to manageable levels. This is because diagnostic constraints are now needed only for the undiagnosed fault pairs of Phase-1. Also, there will be a further reduction in the number of diagnostic constraints due to diagnostically independent fault pairs that could be present. We can also drop the detection constraints as we have started with a detection test set that detects all targeted faults.

There is an additional benefit of the test set obtained by the two-phase approach [16]. For all good chips, testing can be stopped at the end of Phase-1 detection test set, which is of minimal size. Only for bad chips whose number will depend on the yield, we need to apply the remaining tests for diagnosis.

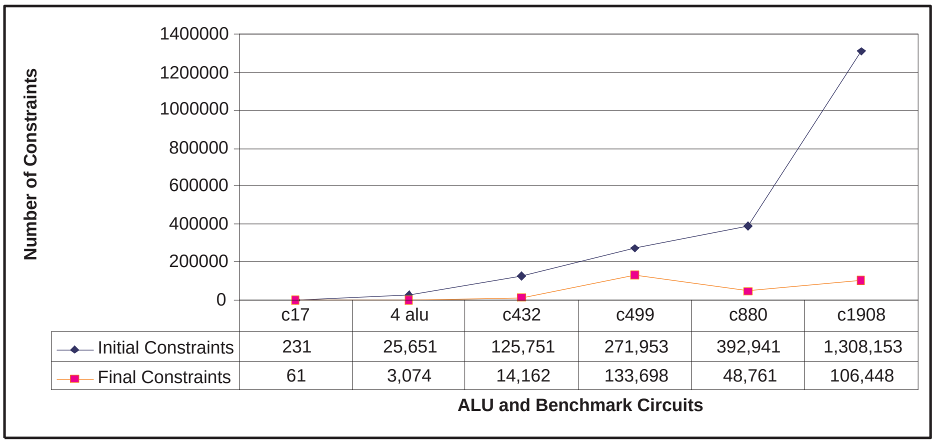

Figure 5. Constraint set sizes

5. Results

In our experiments we have used the ATPG ATALANTA [13] and fault simulator HOPE [14]. We have used AMPL package for ILP formulation.

Results of Phase-1 are given in Table 3. First column lists the names of the ISCAS85 circuits. The next column gives the number of faults in the target fault list. These faults are equivalence collapsed single stuck-at faults, excluding the ones that were identified as redundant or were aborted by the ATPG program. We have used the minimal detection test sets obtained using the primal-dual ILP algorithm [17]. The primal-dual ILP algorithm creates unoptimized test sets which essentially consist of N-detect tests, and then minimizes them to give the minimal detection test sets. The sizes of the unoptimized and minimized vector sets are given in columns 3 and 4 of the table. The subsequent columns give the diagnosis statistics of the minimal detection test sets. We say a fault is *uniquely diagnosed* if it has a unique syndrome. On the other hand a fault whose syndrome is shared by other faults is said to be *undiagnosed.* Column 5 gives the number of undiagnosed faults. Faults with identical syndromes are grouped into a single set called an equivalent fault set. Note that such an equivalent fault set is dependent on the vector set used for diagnosis, thus it is called a *Conditional Equivalent Fault Set (CEFS).* The column, $No.\ of\ CEFS$ gives the number of such sets. There is one CEFS for every non-unique syndrome consisting of the undiagnosed faults associated with that syndrome. *Maximum faults per syndrome* give the maximum number of faults associated with a syndrome. *Diagnostic resolution* (DR) defined in [2] gives an average number of faults per syndrome. It is obtained by dividing the total number of faults by the total number of syndromes. These two parameters quantify the effectiveness of diagnosis since DR indicates how well faults are distributed among all syndromes and the Maximum faults per syndrome indicate the worst distribution among all syndromes. The undiagnosed faults obtained in this step are the target faults in Phase-2 of our algorithm.

Table 4 gives the results for Phase-2 in which diagnostic ILP is used to minimize the tests for the undistinguished fault pairs of Phase-1. In this step we have used the unoptimized test sets (excluding the minimal detection tests) of Phase-1. The $No.\ of\ Faults$ here are the undiagnosed faults from Table 3. The next column gives the number of constraints generated during the ILP formulation. It can be seen that the constraint set size is very small even for the larger benchmark circuits like c7552 and c6288. The column *Minimized Vectors* gives the result of the diagnostic ILP. These vectors combined with the minimal detection vectors of Phase-1 constitute the complete diagnostic test set. The last column gives the CPU time for the diagnostic ILP runs. It is evident that the complexity of the diagnostic ILP is greatly reduced. All CPU times are for a SUN Fire 280R 900MHz Dual Core machine. For cases in which the ILP complexity is high, reduced-complexity ILP variations described in [10] can be used.

Table 5 gives the results and statistics of the fault dictionary obtained by using the complete diagnostic test set. The *total diagnostic vectors* are the combined vector sets from Phase-1 and 2. Notice that these test sets are just a little bigger than the minimal detection test sets of Table 3. Thus failed chips can be diagnosed very quickly as the detection tests would have already been applied during testing. Column 3 gives the number of faults in the target fault list. Column 4 gives the number of uniquely diagnosed faults. The remaining columns have similar meaning to that of Table 4. It can be seen that there is an improvement in the diagnostic resolution from that of Phase-1 due to the diagnosis vectors from Phase-2.

The unoptimized test sets used in our experiments are essentially N-detect tests. It should be noted here that using an unoptimized test set consisting of diagnostic ATPG vectors [18] will be more effective in achieving a good diagnostic resolution, as these vectors are generated for the sole purpose of distinguishing pairs of faults.

Table 6 gives a comparison between the two-phase minimization and another test compaction algorithm for pass-fail dictionary [8]. For both algorithms an initial unoptimized set of 1024 random vectors is used. The authors of [8] measure the diagnostic effectiveness of the compacted test set in terms of number of undiagnosed fault pairs. The pass-fail dictionaries have inherently lower resolution than the full-response dictionaries. Thus, there may not be a one-to-one comparison between the two results. However, we still notice the compactness of the diagnostic test sets and the computing efficiency of the two-phase method.

6. Conclusion

We have presented an integer linear program (ILP) formulation for compaction of the test set used in full-response dictionary based fault diagnosis. The compaction is carried out without any compromise on the diagnostic resolution of the initial test set. The newly defined generalized independence relation between pairs of faults is very effective in reducing the number of constraints in the diagnostic ILP. Finally we have proposed a two-phase approach for generating a minimal diagnostic test set. The diagnostic test sets obtained are very small because of which there can be a significant reduction in the fault dictionary size and also the diagnosis time. Also, the minimized fault dictionary can be further compacted by other compaction techniques that employ encoding of the data in the dictionary.

Recent work on N-model test minimization [19] shows how a single detection table can be constructed for tests of multiple fault models. One may use that idea to create a fault dictionary for multiple fault models and then use the two-phase approach to minimize the diagnostic vector set. Such a fault dictionary would be more effective in diagnosing real defects.

References

[1] M. Abramovici and M. A. Breuer, “Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis,” *IEEE Trans. Computers*, vol. C-29, no. 6, pp. 451–460, June 1980.

[2] V. D. Agrawal, D. H. Baik, Y. C. Kim, and K. K. Saluja, “Exclusive Test and its Applications to Fault Diagnosis,” in *Proc. 16th International Conf. VLSI Design*, Jan. 2003, pp. 143–148.

[3] S. B. Akers, C. Joseph, and B. Krishnamurthy, “On the Role of Independent Fault Sets in the Generation of Minimal Test Sets,” in *Proc. International Test Conf.*, 1987, pp. 1100–1107.

[4] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits.* Boston: Springer, 2000.

[5] B. Chess and T. Larrabee, “Creating Small Fault Dictionaries,” *IEEE Trans. Computer-Aided Design*, vol. 18, no. 3, pp. 346–356, Mar. 1999.

[6] P. Drineas and Y. Makris, “Test Sequence Compaction Through Integer Programming,” in *Proc. International Conf. Computer Design*, 2003, pp. 380–386.

[7] P. F. Flores, H. C. Neto, and J. P. Marques Silva, “An Exact Solution to the Minimum Size Test Pattern Problem,” *ACM Trans. Design Automation of Electronic Systems*, vol. 6, no. 4, pp. 629–644, Oct. 2001.

[8] Y. Higami, K. K. Saluja, H. Takahashi, S. Kobayashi, and Y. Takamatsu, “Compaction of Pass/Fail-based Diagnostic Test Vectors for Combinational and Sequential Circuits,” in *Proc. ASPDAC*, 2006, pp. 75–80.

[9] K. R. Kantipudi and V. D. Agrawal, “On the Size and Generation of Minimal N-Detection Tests,” in *Proc. 19th Int. Conf. VLSI Design*, Jan. 2006, pp. 425–430.

[10] K. R. Kantipudi and V. D. Agrawal, “A Reduced Complexity Algorithm for Minimizing N-Detect Tests,” in *Proc. 20th Int. Conf. VLSI Design*, Jan. 2007, pp. 492–497.

[11] D. Lavo, B. Chess, T. Larrabee, and F. J. Ferguson, “Diagnosing Realistic Bridging Faults with Single Stuck-at Information,” *IEEE Trans. Computer-Aided Design*, vol. 17, no. 3, pp. 255–268, Mar. 1998.

[12] D. Lavo and T. Larrabee, “Making Cause-Effect Cost Effective: Low-Resolution Fault Dictionaries,” in *Proc. International Test Conf.*, 2001, pp. 278–286.

Table 3. Phase-1: Diagnosis with minimal detection test sets.

Circuit	No. of faults	Primal-dual ILP algorithm [17]		Minimal detection test diagnostic statistics			
		No. of unoptimized vectors	No. of minimized vectors	No. of undiagnosed faults	No. of CEFS	Maximum faults per syndrome	Diagnostic resolution
4b ALU	227	270	12	43	19	4	1.118
c17	22	32	4	6	3	2	1.158
c432	520	2036	30	153	68	9	1.195
c499	750	705	52	28	12	3	1.023
c880	942	1384	24	172	83	4	1.1
c1355	1566	903	84	1172	531	5	1.693
c1908	1870	1479	107	543	248	17	1.187
c2670	2630	4200	70	833	316	11	1.245
c3540	3291	3969	95	761	313	8	1.158
c5315	5291	1295	63	1185	527	8	1.142
c6288	7710	361	16	2416	1122	6	1.202
c7552	7419	4924	122	1966	891	7	1.17

Table 4. Phase-2: Diagnostic ILP Minimization.

Circuit	Number of Unoptimized Vectors	No. of Faults	No. of Constraints	Minimized Vectors	CPU s
4b ALU	258	43	30	6	1.36
c17	28	6	3	2	1.07
c432	2006	153	101	21	3.03
c499	652	28	10	2	1.09
c880	1358	172	41	7	2.74
c1355	1131	1172	12	2	2.13
c1908	819	543	186	21	3.16
c2670	4058	833	383	51	5.29
c3540	3874	761	146	27	8.45
c5315	1232	1185	405	42	15.35
c6288	345	2416	534	12	50.13
c7552	4802	1966	196	31	9.35

[13] H. K. Lee and D. S. Ha, *On the Generation of Test Patterns for Combinational Circuits*. Tech. Report 12-93, Dept. of Elec. Eng., Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1993.

[14] H. K. Lee and D. S. Ha, "HOPE: An Effcient Parallel Fault Simulator for Synchronous Sequential Circuits," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 9, pp. 1048–1058, Sept. 1996.

[15] I. Pomeranz and S. M. Reddy, "On the Generation of Small Dictionaries for Fault Location," in *Proc. International Conf. Computer-Aided Design*, 1992, pp. 272–278.

[16] M. A. Shukoor, "Fault Detection and Diagnostic Test Set Minimization," Master's thesis, Auburn University, ECE Department, May 2009.

[17] M. A. Shukoor and V. D. Agrawal, "A Primal-Dual Solution to the Minimal Test Generation Problem," in *Proc. 12th VLSI Design and Test Symp.*, 2008, pp. 169–179.

[18] A. Veneris, R. Chang, M. S. Abadir, and M. Amiri, "Fault equivalence and diagnostic test generation using ATPG," in *Proc. International Symp. Circuits and Systems*, 2004, pp. 221–224.

[19] N. Yogi and V. D. Agrawal, "N-Model Tests for VLSI Circuits," in *Proc. 40th Southeastern Symp. System Theory*, 2008, pp. 242–246.

Table 5. Diagnosis with complete diagnostic test set.

1 Circuit	2 Total Diag. Vectors	3 No. of Faults	4 Uniquely Diag. Faults	5 No. of CEFS	6 Undiag. faults (3 - 4)	7 No. of Synd. (4 + 5)	8 Maximum Faults per Syndrome	9 Diagnostic Resolution (3 / 7)
4b ALU	18	227	227	0	0	227	1	1
c17	6	22	22	0	0	22	1	1
c432	51	520	488	16	32	504	2	1.032
c499	54	750	726	12	24	738	2	1.016
c880	33	942	832	55	110	887	2	1.062
c1355	86	1566	397	532	1169	929	3	1.686
c1908	127	1870	1380	238	490	1618	8	1.156
c2670	121	2630	2027	263	603	2290	11	1.149
c3540	122	3291	2720	234	571	3033	8	1.085
c5315	105	5291	4496	381	795	4877	4	1.085
c6288	28	7710	5690	1009	2020	6699	3	1.151
c7552	153	7419	5598	848	1821	6446	7	1.151

Table 6. Two-phase minimization versus previous work [8].

Circuit	Pass-fail dictionary compaction [8]				Two-phase approach (this work)			
	Fault coverage %	Minim. vectors	Undist. fault pairs	CPU* s	Fault coverage %	Minim. Vectors	Undist. Fault pairs	CPU** s
c432	97.52	68	93	0.1	98.66	54	15	0.94
c499	-	-	-	-	98.95	54	12	0.39
c880	97.52	63	104	0.2	97.56	42	64	2.56
c1355	98.57	88	878	0.8	98.6	80	766	0.34
c1908	94.12	139	1208	2.1	95.69	101	399	0.49
c2670	84.4	79	1838	2.8	84.24	69	449	8.45
c3540	94.49	205	1585	10.6	94.52	135	590	17.26
c5315	98.83	188	1579	15.4	98.62	123	472	25.03
c6288	99.56	37	4491	1659	99.56	17	1013	337.89
c7552	91.97	198	4438	33.8	92.32	128	1289	18.57

*Pentium IV 2.6 GHz machine **SUN Fire 280R, 900 MHz Dual Core machine